Experiences with Implementing RINA - Recursive Inter Network Architecture

Miguel Ponce de Leon, Eduard Grasa, Eleni Trouva, Patrick Phelan,

FIA Poznan 2011: Session 2.4 - Architectures
25 Oct 2011
What is this presentation about?

• Highlight that the current Internet architecture - based on functional layering and TCP/IP is not complete!
• Highlight that past efforts (70s, 80s) provided a more complete architecture with less fundamental limitations!
• Introduce a new architecture that provides a structure that overcomes these limitations: RINA
• Provide an example implementation
Wait a minute! Who am I?

• The goals of the talk look a bit ambitious, don’t they?

• Of course much work has been done by John Day, Louis Pouzin, and other networking pioneers.

• I’m just learning from them, with them and want to understand the issues more, don’t you?
Who is John Day and what is RINA?

- RINA: Recursive InterNetwork Architecture – is an Internet architecture proposed to resolve the challenges of today's internet.

–Patterns in Network Architecture: A Return to Fundamentals by John Day, Published by Prentice Hall ISBN 0132252422
Networking is IPC

• RINA is based on the principle that “Networking is a distributed application, specialised to do Inter Process Communication”
  – All entities in computer systems are processes.
  – They communicate with each other using IPC services.

• IPC within one system is easy: local access control, shared memory, message passing, local queues, …

• IPC between two or more systems is more complicated: need error control, flow control, multiplexing, distributed access control, …
  – Need a distributed application that provides IPC over a certain scope (a direct link, a network, an internetwork).
IPC over different scopes

- Different examples of IPC between application process A and application process B.
  - The mechanisms for doing IPC in the different cases are the same, they just need a different configuration.

- It is a repeating structure: distributed applications doing IPC at a bigger scope (network) use the services of distributed IPC at smaller scopes (link)
  - Recursive structure of distributed applications that do IPC!
Distributed IPC Facility (DIF)

- IPC process. Application Process that does IPC.
- DIF. Distributed IPC Facility; set of IPC processes that cooperate to provide distributed IPC.
- The structure can accommodate as many levels of DIFs as required by the network designer.

DIFs provide the following service (interface):
- Allocate a flow to the destination application with a certain QoS
- Write data to the flow
- Read data from the flow
- Deallocate the flow resources
RINA Design Principles

• Principle 1: Recursive Layers: the ends are relative

• Principle 2: Separate mechanism from policy: design for multiple functions.

• Principle 3: Complete naming and addressing architecture: support for mobility and multihoming.

• Principle 4: Security by design.

• Principle 5: Separation of port allocation from synchronisation in the transport protocol.

• Principle 6: Integrating connectionless and connection-oriented networking.
Principle 1: Recursive Layers: the ends are relative

- Functional Layering the perfect world: This time it’s not just TCP/IP
  - CYCLADES, DECNET, XNS and OSI also used functional layering as an organizing principle

- Functional layering: Each layer is a black box, performing a single function not repeated by the other layers of the stack.
  - Each layer provides a different service to the layers above
  - Fixed number of layers (5, 7)
Principle 1: Recursive Layers: the ends are relative

- Functional Layering the Issues

- Functions are repeated at different layers
  - Error and flow control on data link and transport
  - Routing in several layers (spanning tree at layer 2, OSPF at layer 3)

- Protocols that do not match a layer strictly
  - MPLS (Layer 2.5? What is this layer?)

- Layer violations
  - TCP pseudo-header (requires data from the IP layer)

- What we are seeing today are: repeated functions, protocols that do not fit in layers, broken layering, …
  - Doesn’t seem like an upheld principle for the current Internet.
Principle 1: Recursive Layers: the ends are relative

- Functional Layering the Issues

- In fact from a single link to the whole Internet, there is NO layering which means that distributed functions have to manage the complete granularity of resource allocation at once.

- No layering means protocols have to operate effectively over different physical media and provide an efficient service to different applications at the same time.

- Layers are supposed to provide a divide-and-conquer approach, by isolating shared state of different scope (link, network, network of networks, …).

- There should be no separation between networks and applications.

- Which leads us to the principle that: “networking is IPC and only IPC”
  - As all the entities executing in computing systems are processes, networking can be seen as distributed inter-process communication
The First RINA prototype is being build on top of UDP/IP.

Based on Java, but applications don’t need to use Java to interact with it.

The prototype uses underlying IP Networks as if they were the physical media.
Basic Cloud deployment scenarios

- **Service Provider**: Internal infrastructure
- **Infrastructure Provider**: Bursting internal clouds
- **Federated clouds**: Service Provider
- **Multi-clouds**: Broker
- **Source**: OPTIMIS project
RINA incremental deployment strategy

- Application Model specification
- Set of Virtual Computing Resources
- Storage Resources
- Network topology
- RINA (DIFs)
- System Portability VM Migration
- Service Migration
- Secure and Dynamic network configuration
- Data Portability and Clustering

RINA-enabled experimental network between organisations

Source: ATOS
Example deployment of the prototype

- The picture shows two hosts hosting applications A and B (A and B are application names).
- The IPC processes in each DIF are shown with its address (not its application name). As they are internal to each DIF, they can be repeated between DIFs.
- DIFs have a name (it is its distributed application name).
IPC Process interface

• Provides at least the 4 basic operations:
  – `int portId = allocate (Source AP name, destination AP name, QoS params)`
  – `byte[] pdu = read(portId)`
  – `write(byte[] pdu, portId)`
  – `dallocate(portId)`

• Example (referring to slide 15)
  – App A would invoke `allocate(A,B,[params])`
  – IPC Process 1 in DIF AA would:
    • Query its local directory to find that B is reachable through IPC Process 4
    • Map the QoS params to one of the QoS cubes (categories) available in DIF AA
    • Instantiate the Data Transfer Protocol with the right policies to comply with the selected QoS cube
    • Send a “Create Flow” request to IPC Process 4 (using the CDAP Protocol)
    • Upon successful reception of the “Create Flow” response, it notifies App A, passing it the `portId`
    • App A can start reading and writing PDUs
Prototype limitations (= research opportunities)!

- Will just support two QoS cubes
  - 1 equivalent to TCP and 1 equivalent to UDP
  - Research topic: provide policies that provide support for different QoS cubes

- Has to work on top of UDP/IP
  - Research topic: Improve it to run on top of Ethernet, implement the right policies
  - Research topic: Prototype on top of wireless (WiFi), implement the right policies

- Addressing is just simple enumeration
  - Research topic: Topological addressing schemes to optimize routing

- Congestion control not implemented
  - Research topic: Strategies for congestion control within a DIF. Influence of congestion control
Prototype limitations (= research opportunities)!

• Rudimentary resource allocation
  – Research topic: Explore different resource allocation strategies (e.g. to dimension the different queues, to design the different schedulers, …). Exploit the unification of the connection and connectionless approaches.

• Authentication/Authorization not implemented
  – Research topic: Plug-in different authentication/authorization schemes and study their pros&cons, suitability to different apps, …

• Rudimentary management capabilities
  – The application protocol (management protocol) is there: CDAP, and a rudimentary DMS (DIF management system) is expected, mainly to manage the instantiation of new DIFs
  – Research topic: Complete DMS, what are the required objects, how to design the RIB (Resource Information Base, equivalent to the MIB), exploit the autonomic properties of the DIF for management, …

• And more open areas yet (whatevercast, DIF structures, application naming structures, …)
Implementing RINA

- A detailed overview of RINA innovations and features can be found the Pouzin Society (PSOC) website [http://www.pouzinsociety.org/].
  - Developing open implementation of RINA within OpenTinos.
  - Experimental Sandbox / Testbed.
  - OpenTinos has it origins as a prototype in the EU FP7 research project ICT 4WARD,
Miguel Ponce de Leon,
Chief Technologist,
TSSG, WIT
miguelpdl@tssg.org
+353 51 302952 (w)

www.tssg.org/people/miguelpdl
twitter: miguelpdl